

# EOF-Using EOF-Logging

## Overview

You can turn SQL logging on when launching by using `-DEOAdaptorDebugEnabled=true` or `EOAdaptorDebugEnabled YES`. However, this can slam out a ton of SQL. While this is useful for some debugging it is way too much to sift through when trying to analyze a particular problem.

You can use this method to selectively enable and disable SQL logging:

```
public static void logSQL(boolean shouldLog) {
    if (shouldLog) {
        NSLog.allowDebugLoggingForGroups(NSLog.DebugGroupSQLGeneration |
                                         NSLog.DebugGroupDatabaseAccess |
                                         NSLog.DebugGroupEnterpriseObjects);
    } else {
        NSLog.refuseDebugLoggingForGroups(NSLog.DebugGroupSQLGeneration |
                                          NSLog.DebugGroupDatabaseAccess |
                                          NSLog.DebugGroupEnterpriseObjects);
    }
}
```

This can be useful for such things as determining what fetches are performed when a page is rendered:

```
public void appendToResponse(WOResponse aResponse,
                             WOContext aContext) {
    logSQL(true);
    super.appendToResponse(aResponse, aContext);
    logSQL(false);
}
```

**WO:** Here is an [awk script](#) that takes an a WOApplication log which contains EOAdaptorLog info as input, picks out the SQL statements and their bindings, and spits out a CSV formatted log of those statements, with binding values filled in as literals, along with rowcount and execution time in seconds. Those statements can most of the time be used manually against a database for testing purposes. You need gawk installed for this.